Short manual for TEXworks

Alain Delmotte

December 6, 2008

Contents

1 Introduction 2

2 Installation 2

2.1 Under Windows . . . 3

2.2 Under Linux . . . 3

2.3 Under Mac OS . . . 3

3 First approach 4

3.1 Short description of the interface . . . 4

3.2 Creating un document . . . 4

3.2.1 Writing the document . . . 4

3.2.2 Typesetting the document and viewing it . . . 5

3.2.3 The work of LATEX . . . 5

3.3 And when errors occur . . . 7

3.4 Changing a little bit TEXworks parameters for convenience . . . 8

4 Going further with TEXworks 8

4.1 Spell-check . . . 8

4.2 Toggle source/view . . . 8

4.3 Creating a document from a template . . . 8

4.4 Creating a project using several source files . . . 9

4.5 Other special commands . . . 9

4.6 Classical editing tools and other . . . 9

4.7 Search and replace . . . 10

4.7.1 Classical actions . . . 10

4.7.2 Regular expressions . . . 11

4.8 Auto-completion . . . 12

4.9 Formatting the source for legibility . . . 14

4.10 Showing the tags . . . 14

4.11 Organising the windows . . . 14

5 Annexes **16**
5.1 Keyboard short-cuts 16
5.2 Regular expressions 17
5.3 Roots for completion 18
5.4 Compiling TEXworks under Linux 29

References **31**

1 Introduction

TEXworks is a project to create a text editor oriented for LATEX. Instead to create a new sophisticated editor, fully equipped by multiple tool-bars to meet any need, TEXworks on the contrary try to provide a simple editor, offering at first sight only some limited tools for text editing as well as one button and a menu to typeset a LATEXtext.

The idea to create the editor came to Jonathan Kew, the initiator in charge of the project, after a long reflection on the reasons why potential users tend to keep away from LATEX, as well as from the success of the TEXshop editor on the Mac.

Finally the goal was also to provide the same editor on many operating systems; TEXworks runs for the moment on Linux, Mac OS as well as Windows. The interface is always the same and the program offers the same possibilities.

The first section explain how to install the software. In the second one describes the interface and one creates a first document. In the third section one sees the advanced tools proposed by TEXworks; one should read this section only after mastering the basic working system of TEXworks. The tools presented allow to be much more effective. Finally the last section, as annexes, provides lists for the keyboard short-cuts, the commands known as regular expression, as well as the roots for auto-completion. A bibliography closes this introduction.

2 Installation

TEXworks is only a text editor; to be able to create documents with LATEX and to typeset them to PDF, we need what is called a TEX distribution. It is a bunch of programmes and other necessary files which will be automatically called by TEXworks during its work. So one needs to install a distribution; we will do that “before” to start TEXworks for the first time, this way TEXworks will automatically find what it needs.

One can use **TeXlive** (`http://www.tug.org/texlive/`) which is a combination of teTeX, macTeX and XEmTeX and which is available for the three operating systems. The last available version is TeXlive 2008.

For Linux: every Linux distribution has a LATEX distribution; but it could not be installed by default and one has to use the installation tools to do that. Except TeXlive, one can use **teTeX** (`http://www.tug.org/teTeX/` on which TeXlive is based.

For the mac: **MacTeX**, new distribution based on gwTeX and XeTeX. See also on the WEB at `http://www.tug.org/mactex/`.

For Windows: an often used distribution is **MiKTeX** (`http://www.miktex.org/`). MiKTeX has an update programme, which has also been ported to Linux. One can also use the XEmTeX distribution (`http://www.xemtex.org/`).

For TEXworks, download the archived programmes from the TEXworks web site: `htp://tug.org/texworks/`; one can find binaries for Mac and Windows at: `http://code.google.com/p/texworks/downloads/list`.

One has to get: `TeXworksW32-r238.zip` (or `TeXworksOSX-r238.zip` for the Mac), the programme itself, as well as `TeXworksW32-libs-20080714.zip` (and for the Mac `TeXworksOSX-libs-20080714.zip`), some needed library files [1].

2.1 Under Windows

We create a folder, for example `C:\Program files\TeXworks`, and un-archive the downloaded files in it. Create, on the desktop or in the quick launch bar a short-cut for the `TeXworks.exe` file.

When you start the program the first time, it creates in the folder associated to your account (`C:\Documents and Settings\<vourname>\`) a folder named `TeXworks` which will contain some folders for the auto-completion, configuration, dictionaries, templates, and localisation files – but we will see later. [2]

NB. Up to the version used here, the fact that the name of the main folder of the user, in "Documents and Settings", has accented characters, in fact any non-ASCII character, will prevent the use of the spell-checker and of the automatic shift from the source to the view and back with positionning at the same place.

2.2 Under Linux

Under Linux, most of the time one build the programme by compiling; see the annexe at section 5.4. Once the compilation is done, start TEXworks.

A folder `.texworks` will be created in your `home` directory.

2.3 Under Mac OS

Put the libraries and frameworks in the indicated folders, unless you already have them installed.

On Mac OS X, the TEXworks resources folder will be created in your `Library` folder, inside your `home` directory. Preferences are stored in `~/Library/Preferences/org.tug.TeXworks.plist` which you can delete if you ever suspect it is causing problems.

[1] available versions at the time of writing this manual.

[2] TEXworks will save its preferences in the register: `\HKEY_CURRENT_USER\Software\TUG\TeXworks`. If they are erased, they will be recreated with default values at the next use.

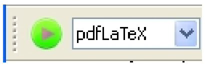
3 First approach

Let's now see how to do a first document: for this one has to type it in the editor window of TEXworks. LATEX is not a WYSIWIG software [3], so you'll have to type the text and the instructions for formatting it and you'll see the result only after "typesetting" the text.

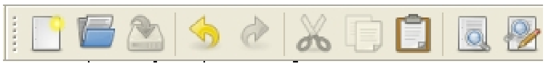
This looks a little bit dry, but one very quickly gets used to it and this is worth the result.

3.1 Short description of the interface

When one opens the editor, it shows a very skinny(?) interface: a title bar, a menu bar, two small tool-bars, a great typing zone and, at the bottom, a status bar.



The first tool-bar has a button to typeset and an unfolding menu to choose the format for typesetting (we'll choose `pdfLaTeX`). Knowing that the keyboard short-cut for typesetting is `CTRL+T` and that we almost never change the format, one could even not show this tool-bar. Further the selection can be done through the **Composition** menu.



The second only provides classical buttons: New document, Open, Save | Undo, Redo | Cut, Copy, Paste | Search, Replace.

3.2 Creating un document

3.2.1 Writing the document

Let's create now the first document! Enter exactly the following text! [4]

```
\documentclass{article}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{geometry}
\geometry{a4paper}

\usepackage[francais]{babel}

\title{Premier document}
\author{Un TeXnicien}
\date{}
```

[3] *What You See Is What You Get*

[4] It is intentionally in French to show some possibilities!

```
\begin{document}
\maketitle

Voici un texte accentué en français!

\end{document}
```

One have to save the document, putting it in a folder, which we specially create for the tests; the name of the document file, for example `first.tex`, should have a `.tex` extension.

3.2.2 Typesetting the document and viewing it

Next we start typesetting [5] by clicking the green button ▶ or by `CTR+T`.

A new panel opens between the typing area and the status bar, it is the *output panel*; there goes everything LaTeX is doing when it works; when it is finished and if there is no error, this panel disappears and a new window appears next to the first one; in this window one can see a page with a title "Premier document" followed by the name of the author "Un TeXnicien", both centred, a text "Voici un texte accentué en français !" and at the bottom centre a page number.

Remark that the mouse cursor is like a magnifier! If you push the left button of the mouse you can see the text under the magnifier much bigger (it is a magnifier, isn't); you can move the magnifier and so inspect the text in details.

To go back to the source, you can just click in its window or better, you'll see by using, do `CTRL+'`. This last short-cut is a toggle between the two windows. [6]

See also **??** the automatic move at a specified location of the view from the source or the inverse.

3.2.3 The work of LaTeX

Let's now shortly analyse the result to understand what LaTeX did and why. Introductions and full tutorials can be found on Internet; see for example *lshort* which should be in the installed LaTeX distribution or which can be downloaded from the net: do a search on CTAN. [7]

We ask to create a document of the class: *article*, it is the global layout of the document.

[5] We also use the words compilation and compile for the same action, indeed LaTeX works on the source file to produce a .pdf output, so there is a compilation.

[6] Under Windows one can use `Alt+Tab` to go to the last window opened before the one in use.

[7] *Comprehensive TeX Archives Network*, it is a network of mirror deposits of the central CTAN, one can find there everything about TeX and LaTeX`http://www.ctan.org`

Next we say that the input document (the source) is saved with the Unicode format utf-8 and that it will then contain characters which are not present in the standard ASCII without accents. We also want to use an output format T1; we also want an *A4* document and not the us *letter* format. Finally we make it clear that the typography should follow the French rules. Those general instructions for the work are done by packages called with options.

Lastly we finish the declaration part of the document, the preamble, giving the title, the author and the date of the document.

Next comes the body of the document, between `\begin{document}` and `\end{document}`. It is here where everything which will appear in the document will be.

Let's do some experiments to show the effect of these instructions. For this we put an % in front of the instructions; the % and everything after it will be considered as comment, that part will then be ignored by LaTeX. [8]

Comment the lines loading the packages one after the other (`\usepackage[]{}`). When you comment the French package, the typesetting will stop (for LaTeX there is an error due to the previous work), just type `[Enter]` to go on. Observe in details the result, for example with the magnifier the exclamation mark and the surrounding text; see also if all characters are present.

After these experiments, let's modify the text as follows:

```
\begin{document}
\maketitle
\tableofcontents

\section{Petite démonstration}

Voici un texte accentué en français!
Suite du texte entré après avoir fait un retour chariot. Dans l'éditeur
on peut demander un passage à la ligne du texte saisi; mais le numéro de
ligne n'est incrémenté que par un retour chariot.

Nouvelle ligne en passant une ligne dans la source: c'est la manière
d'indiquer un changement de paragraphe.

\end{document}
```

Redo the previous experiments and observe the changes which appear.

Note that entering only one carriage return doesn't create a new paragraph. In LaTeX one has to have an empty line for that. In TeXworks the line number of the source (on the right in the status bar) numbers the lines created with carriage return, not the wrapped lines.

[8] Remark that the comments are coloured in red, so we see them well.

3.3 And when errors occur

When you create a document for typesetting with LaTeX, you can avoid making mistakes: forgetting a closing brace or a `\end{}` command to close an environment, no command to switch to the mathematical mode but use of mathematical commands,... When you compile, when there is an error, LaTeX stops, this stop is visible by the stop of the scrolling actions in the output panel, an error message is displayed and LaTeX waits an instruction to know what it should do: one sees the *typing cursor* in the line between the output panel and the status bar.

Note that we can stop the typesetting, the green button changed to a red one with a white cross ✖, by clicking on that button or else with `[CTR+T]`. The output panel is still visible and so one can still see the error message.

The error message is on many lines, like this:

```
! Undefined control sequence.
l.168 ... fermante ou d'une commande \veb
                                         +\end{}+ de fermeture d'un...

?
```

LaTeX says that it doesn't recognize the command name, sometimes suggests to see the manual or to type `h` (plus `[Return]`) for help, points to the line number (here 168) and the place of the error at the cut of the line (here at `\veb`) and finally with the question mark shows that it waits for an action from us.

There are different possible actions:

- type `[Return]` and ask to continue as if nothing happened; sometimes this allows to finish compiling, but there will be an error in the result;
- type `h[Return]` to ask for help; this help in not always clearer than the error message, but often this gives a clue;
- type `i[Return]` to tell LaTeX that we will propose a replacement text, enter it followed by `[Return]`, it will be used, beginning at the level of the error, but you should correct the source afterwards; there is no correction in the source during compilation;
- type `x[Return]` to stop compilation.

One should note that sometimes an error appears far from its real position, for example opening an environment but not closing it, LaTeX doesn't see the error before it encounters another end of environment without closing of the first one; it is often the `\end{document}` which shows that another environment was not closed!

3.4 Changing a little bit TEXworks parameters for convenience

If the default font of the editor doesn't suit you, it is possible to change it by ***Format / Font...*** and selection in the dialogue box which appears. This change will only be temporary, one gets back the default font if one closes TEXworks and opens it again.

From the ***Typeset*** menu or from the unfolding menu of the ***Typesetting tool bar*** one can change the compilation format. Again this change will only be temporary.

To have a permanent change, one has to change the *preferences* through the ***Edition / Preferences...*** menu, panel ***Editor*** for the font and panel ***Typesetting***, at the bottom, for the default format (let's choose `pdflatex` for this one).

4 Going further with TEXworks

When you'll have some practise with TEXworks, you'll find the need for more effective tools. Many tools exist in TEXworks. We are going to see them now.

4.1 Spell-check

One can ask to have automatic spell-check during typing by ***Edit / Spelling / <selection of the language>***: as an example fr-FR for French.

During typing, if there is an error, the word is underlined by a red underline. A right-click on the word opens a contextual menu in which there are some replacement suggestions. Click on the desired word to make the replacement.

Before to use the spell-check, one has to install dictionaries in the right folder of TEXworks: `C:\Documents and Settings\<yourname>\TeXworks\dictionaries`.

One can use the available dictionaries for Open office and other free software [9]; for example, if you have Thunderbird with spell-check, you can copy the `.aff` and `.dic` files. it is possible to ask TEXworks to use by default a dictionary by ***Edit / Preferences.../ Editor*** option ***Spell-check language:***.

4.2 Toggle source/view

When one reads the document in the view and one sees an error, it is interesting to go immediately at the same place in the source. To do that it is enough to click at the selected place in the view holding `[Ctrl]` down (`[Ctrl+Click]`, one jumps to the searched location in the source.

De same way, if one has already shown the view and one is back in the source, one can go back to the view at a specific location by `[Ctrl+Click]`.

4.3 Creating a document from a template

The documents that we create use most of the time the same instructions in the preamble, one loads the same packages, one specifies the same page layouts, one also specifies

[9] see for example at `http://lingucomponent.openoffice.org/download_dictionary.html`

personal headers and footers,...

One can use predefined templates or create one's own having all those pre-requirements.

Use ***File / New from template...*** or `Ctrl+Shift+N`. A dialogue box opens to allow selection of the template. After selection and `OK` a document is created and one can start to work.

If one desires to create a more personal template, one has just to create this document with everything one always wants to find (and perhaps marking places to fill in) and save it as a `.tex` files in the TEXworks folder or a possible sub-folder of this one `C:\Documents and Settings\<yourname>\TeXworks\templates` [10].

4.4 Creating a project using several source files

When the source becomes long, it is sometimes difficult to move in it. It is then interesting to split the source in different smaller files: one file will be the main document, central, with the preamble, the `document` environment, as well as calls to the "sub-documents" [11].

But there might be a problem if, being in a sub-document, on start typesetting/compilation; as there is no preamble nor `document` environment there is an immediate error stop.

To tell TEXworks that it should typeset the main document one adds at the very beginning of the sub-document the instruction:

```
% !TeX root = chemin/au/fichier_principal.tex
```

for example:

```
% !TeX root = manual.tex
```

if the main file is in the same folder, its name is enough, as in the above example. Also remark the use of the slash "`/`" and not the backslash "`\`" used under Windows to separate the levels of folders.

4.5 Other special commands

The same way one can manage two other aspects of the compilation. TEXworks uses, by default, the "utf8" encoding, but some files could be saved in another format. To ask another encoding for a specific file one can put at the beginning of this file:

`% !TeX encoding = latin1` : another encoding often used.

If one wants to compile a file by another programme than the default TEX or LATEX, put at the beginning of the file:

`% !TeX program = the_programme`.

4.6 Classical editing tools and other

TEXworks has the classical editing tools as the clipboard; therefore one can select, cut/copy and paste a piece of text.

[10] case of Windows, see section 2 Installation.

[11] calls by the commands `\input{}` or `\input{}`, see LATEX manuals for more information.

One can select with the mouse by “glide??” on the desired text, one can also “double-click” to select a word. With the keyboard one moves holding the [Shift] key down; one can use only the direction keys with [Shift]; one can also move and select word by word moving left or right holding [Ctrl+Shift] down. The clipboard short-cuts are the short-cuts one finds in all the programmes: [Ctrl-X] to cut, [Ctrl+C] to copy and [Ctrl+V] to paste.

One can easily change the case of a selection – put everything upper case or everything lower case – by ***Edit / Changer case /*** and next depending on the case ***ALL UPPER CASE*** or ***all lower case***.

it is always possible to undo an un-wished action by ***Edit / Undo*** or [Ctrl+Z]; this way one can undo stepwise! The inverse action undo the undo or redo is done by ***Edit / Redo*** or [Ctrl+Maj+Z].

When one prepare a document with LaTeX it is often interesting to prevent compilation of a portion of text to be able to locate an error; one goes on this way piece by piece of good text until one finds a part which creates an error. For this one comments the source.

We have seen that the symbol % shows the beginning of a comment. To comment a big piece of text, it is enough to select it and ask to mark it as comment ***Format / Comment*** or Ctrl+Shift+]. To suppress the comment: ***Format / Uncomment*** or Ctrl+Shift+[.

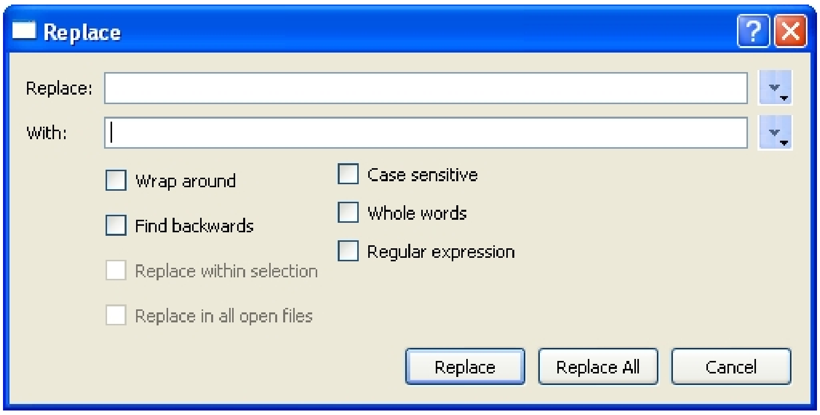
A frequent error is to forget a closing symbol: parenthesis, bracket, square bracket. TeXworks helps with a tool to show the pairs of symbols: when one move over one of these symbols its complementary is briefly highlighted in orange. One can also, when one is inside a block like this, one can ask to select the block by ***Edit / balance delimiters*** or by its short-cut [Ctrl+B]. Thus one immediately sees the scope of the block.

4.7 Search and replace

other classical tools: search and replace text. Of course TeXworks has these possibilities with some more possibilities.

4.7.1 Classical actions

The options of the menu ***Search***: “***Find...***”, “***Find again***”, “***Replace...***” et ***Replace again*** ([Ctrl+F], [Ctrl+G], [Ctrl+R] et [Ctrl+Shift+R] respectively) are classical actions; the first and the third open a dialogue box:



□ Wrap around

□ Find backwards

□ Replace within selection

□ Replace in all open files

□ Case sensitive

□ Whole words

□ Regular expression

There are the usual options: *search forwards* (default), *Find backwards*, *Wrap around* or *Replace within selection*. The following options are also usual *Case sensitive* and *Whole words*.

The option *Replace in all open files* is also a frequent extension, but not as much as the others; this allows, for example, to replace in all the files of a project.

The last option, *Regular expression*, is detailed below.

In the menu ***Search*** there are other interesting options:

- ***Copy to Find***, when one wants to search or replace some text by another one can select the text and send it in the input area *Replace:* of the Replace dialogue box; for only searching without replacement one can also use this option, one then sends to the *Find:* area of the Find dialogue box, but the third option described below is more practical in this case;

- ***Copy to Replace***, one does the same for the replacement text; it is then enough to launch Replace... (`[Ctrl+R]`) and the information are already in the right areas;

- ***Find Selection***, with this it is even not necessary to send to the *Find:* area, next open the dialog box by `[Ctrl+F]`, it is enough to use the (`[Ctrl+H]` command and TEXworks searches the next occurrence of the selection; one can repeat the search by `[Ctrl+G]`;

- ***Show selection***, if one selects some text and after this ones moves somewhere else with mouse lateral slider, entering the `[Ctrl+=]` brings us back to the selected text.

4.7.2 Regular expressions

The regular expressions provide a very powerful tool, but one has to understand them well. There are often used when one writes programmes, as well as in the programmes them-self to manage the content worked upon by the latests. A full manual would be required only to learn them, but we'll give some ideas on how to use. See also the available expressions in section 5.2.

Suppose we have the following text:

```
Voici du texte pour tester les expressions régulières
dans du texte accentué.
Voici du texte pour tester les expressions régulières dans
du texte accentué.
Voici du texte pour tester les expressions régulières. Voici
du texte pour tester les expressions régulières.
truc          truc
tél.: 010-99-99-99
tél.: 00.32.10.99.99.99
tél.: 00/32-10/99.99.99
```

We want to 1) insert an empty line between the paragraphs after "accentué" (paragraph in LaTeX) but not for the three telephone numbers, we also want 2) replace the two tabulations between the two words "truc" of the fourth paragraph, each by three spaces and finally 3) make uniform the telephone numbers replacing the "`-./`" by spaces.

For 1) in the dialogue box ***Replace*** (`[Ctrl+R]`) for *Find:* we put "`>\n<`" [12] and in *With:* "`>\n\n<`". "`\n`" is the code to insert a line feed. One would take care to first select the first four paragraphs and the beginning of the fifth (first telephone number) and to mark off the *Replace within selection*; if this is not done, select the telephone lines and do the reverse action "`>\n\n<`" and "`>\n<`".

For 2) use "`>\t<`" and "`>␣␣␣<`". "`\t`" is the code which represent a tabulation.

For 3) it will be "`>-|\.|/<`" and "`>␣<`". Here "`|`" provides different possibilities; for the point we have used "`\.`" because the dot alone represents any character and we would have replaced all the characters by a space!! we then use a code to get the point.

If on has strings of the same character but of different length (for example 2, 3, 4, 5 characters) and one wants to bring all strings to a string with less characters (for example 2), one can ask to replace the string "`>e{3,5}<`" by "`>ee<`".

If one wants to insert at the beginning of some paragraphs separated or not by an empty line the same string, for example `\noindent␣` or `\item␣`, one can replace "`>\n\n<`" or "`>\n<`" by "`>\n\n\\noindent␣<`" or "`>\n\\noindent␣<`". pay attention, we have double `\` in front of `noindent`!

If it was making sense, we could replace all the letters between "a" and "m" by "$" using "`>[a-m]<`" and "`>$<`".

4.8 Auto-completion

Another tool which rapidly becomes indispensable is auto-completion. Indeed, when one uses LaTeX, ones has continuously to enter codes for, for example, create environments; further in this case one has to remember to close them.

The auto-completion allows to type a string, one could say a keyword, and next typing `[Tab]` the LaTeX command or environment code is automatically created.

[12] here the >< only show the limits of the typed text and should not be typed them-self.

As an example to have "LaTeX", we have to type \LaTeX. This is not difficult, but entering "\" [13] followed by the word "LaTeX" with alternating capitals and lower case letters could become annoying after a while. With auto-completion you just enter latex next [TaB] to get \LaTeX. One just have to take care that there is no *letter* glued in front or after latex.

Other examples, bmin gives:

```
\begin{minipage}{}
•
\end{minipage}
```

the blinking insert location is between the empty pair of brackets where one has to enter the size of the minipage, and xve gives \varepsilon that is ε in mathematical mode. See the section 5.3 for a list of the keywords for auto-completion. Remark the "•" in the minipage environment. It is a place holder which can be reached by [Option+Tab] on the Mac, repeating this one goes forward in the created structure and by [Maj+Option+Tab] one goes backward. [14]

One should remark that if a partial keyword is given and that uses repeatedly [Tab], one can have other completions, generally associated. For example, bali (the b means the beginning of an environment \begin{}) creates the align environment after one [Tab], next align*, and after, in succession, alignat, alignat*, aligned, alignedat, alignedat with option; these last environments have their own code which starts by bali (balis baliat baliats balied baliedat and baliedato.)

Eventually, if you want to create your own short-cuts/keywords, you can add a .txt file in the completion of the TeXworks folder created during the first use (see section **??** on installation.)

The entries in the file should have the following format:

```
bfigo:=\begin{figure}[#INS#]#RET##RET#\end{figure}
\bibliography{#INS#}
```

In the first case, for the figure environment with option, bfigo is the keyword, next comes the assignment := and the definition: write \begin{figure}[] \end{figure} with a line feed after begin (#RET#), create an empty line (second #RET#) and put the cursor between the square brackets (#INS#).

In the second case there will be only \bibliography{} it-self and the cursor between the brackets. The keyword is the instruction itself.

Of course it is always possible to use "•"!

One has to take care to create utf-8 encoded files; they can be created with TeXworks it-self.

[13] in particular on some keyboard, like the French Azerty one, to have \ it is necessary to use [AltGr+<] or [Ctrl+Alt+<]

[14] This doesn't work yet with Windows, under Linux ??

4.9 Formatting the source for legibility

To facilitate legibility of the source, one can use the indentation as the programmers are doing:

```
\begin{itemize}
    \item First element of the list;
    \item second element;
    \item last element:
    \begin{itemize}      % beginning of the sub-list
        \item first sub-element;
        \item second sub-element.
    \end{itemize}
\end{itemize}
```

this increases legibility, but works well only on short lines, without text wrapping; or if one ask not to use text wrapping by ***Format / Wrap lines***.

The command ***Format / Indent*** or the short-cut `Ctrl+]` will indent the line, or the selected lines, moving four characters. One can repeat the process to increase the indent. Each time there is insertion of a tabulation character in the source.

To suppress one indentation: ***Format / Unindent*** or the short-cut `Ctrl+[`.

4.10 Showing the tags

When a document is becoming long and one wants to move to a specific place (a chapter, a section, a sub-section,...) one has to scroll the edition window to find the desired location.

One can also ask to show the structure tags of the document in a panel to the left of the edition panel by ***Window / Show / Balises***; the same instruction allows to mask this panel, as well as a click on the cross in the upper right corner of the panel. A click on the searched level selects the corresponding instruction of the source.

One can even more or less develop the structure if it is very long by a click on the small square with "-" to reduce/close a structure and on the small square with "+" to open it.

The same action is possible in the view by ***Window / Show / Table of contents***, but this only works if one has created structure tags in the PDF file using the package `hyperref`.

4.11 Organising the windows

By default, the edition window opens on the left and the view one, when the corresponding PDF file exists, to the right splitting the screen in two.

One can change the position of the windows in the menu ***Window***. ***Stack*** and ***Side by side*** give the same effect if there is only one document open, if not Stack creates a mosaic with all the windows. The other options allow to place the windows for one's convenience. It is also always possible to re-size and move the windows.

For the view one can change the way it is presented and of course the zoom by ***Actual size***, ***Fit to width*** and ***Fit to window*** using the options of the ***View*** menu; one can also zoom in and out. Short-cuts exist for all these actions and are given next to the option in the menu.

5 Annexes

5.1 Keyboard short-cuts

The use of keyboard short-cuts greatly facilitate typing in and the management of the source and the preview. Their use is much more effective than the use of the mouse on tool-bars' buttons.

You'll find here below the short-cuts for the work in the source and those for the preview

For the work in the source:

Short-cut	Action
`Ctrl+N`	New
`Ctrl+Maj+N`	New from template
`Ctrl+O`	Open
`Ctrl+S`	Save
`Ctrl+Maj+S`	Save as
`Ctrl+W`	Close
`Ctrl+Z`	Undo
`Ctrl+Maj+Z`	Redo
`Ctrl+X`	Cut
`Ctrl+C`	Copy
`Ctrl+V`	Paste
`Ctrl+F`	Search
`Ctrl+G`	Search again
`Ctrl+R`	Replace
`Ctrl+E`	Copy to Find
`Ctrl+Maj+E`	Copy to replace
`Ctrl+L`	Go to line
`Ctrl+H`	Find selection
`Ctrl+=`	Show selection
`Ctrl+A`	Select All
`Ctrl+B`	Balance Delimiters
`Ctrl+]`	Indent
`Ctrl+[`	Unindent
`Ctrl+Maj+]`	Comment
`Ctrl+Maj+[`	Uncomment
`Ctrl+`	Show/hide output panel
`Ctrl+′`	Go to preview
`Tab`	Auto-completion

moves (and selections: `Shift+`)

→	1 character right
`Ctrl+`→	1 word right

←	1 character left
Ctrl+←	1 word left
↑	1 line up
↓	1 line down
PgUp	1 screen up
PgDown	1 screen down
Home	Begin of line
Ctrl+Home	Begin of document
End	End of line
Ctrl+End	End of document

For the work in preview:

Short-cut	Action
Ctrl+N	New
Ctrl+Maj+N	New from template
Ctrl+O	Open
Ctrl+W	Close
Ctrl+Z	Undo
Ctrl+Maj+Z	Redo
Ctrl+X	Cut
Ctrl+C	Copy
Ctrl+V	Paste
Ctrl+RetArr	Erase
Home	First page
End	Last page
PgUp	Previous page
PgDown	Next page
Ctrl+G	Go to page…
Ctrl++	Zoom +
Ctrl+-	Zoom -
Ctrl+1	Actual size
Ctrl+2	Fit to width
Ctrl+3	Fit to window
Ctrl+Maj+F	Full screen
Ctrl+T	Typeset
Ctrl+′	Go to source

5.2 Regular expressions

==============================> to be written

5.3 Roots for completion

We give here the keywords for auto-completion as they are supplied by TeXworks. They are in the files `tw-basic.txt`, `tw-context.txt` (empty) and `tw-latex.txt` from the `TeXworks\Completion` folder.

We give them in three columns: the first two show the keywords, the third the LaTeX code produced. In some cases therre is only the code, this means that one can starts to enter the LaTeX code and try to complete it by `[Tab]`.

During completion, the system inserts line feeds and puts the cursor at the first place where one has to enter information to complete the typing. To represent the line feeds we used $\mathcal{R}$ and $\mathcal{I}$ for the input point.

It is possible to see that the keywords have some coherence, the mathematical variables have a keyword starting with `x`, when they are in a mathematical environment, when they are used alone in the text one adds `d` in front; for example `xa` and `dxa` for `\alpha`, if there is a capital there is a `c`, as `xo` for `\omega` and `xco` for `\Omega`. The keywords for environments start with `b`: `bali` for `align` (`b` for `\begin`). When the environment has possible options, there is one or more `o` added to the base name: `bminp` for `\begin{minipage}{}..` and `bminpo` for `\begin{minipage}[]{}...`

Keywords defined in `tw-basic.txt` (defined in TeX):

xa	\xa	\alpha
xb	\xb	\beta
xch	\xch	\chi
xd	\xd	\delta
xcd	\xcd	\Delta
xe	\xe	\epsilon
xve	\xve	\varepsilon
xet	\xet	\eta
xg	\xg	\gamma
xcg	\xcg	\Gamma
xio	\xio	\iota
xl	\xl	\lambda
xcl	\xcl	\Lambda
xm	\xm	\mu
xn	\xn	\nu
xo	\xo	\omega
xco	\xco	\Omega
xp	\xp	\pi
xcp	\xcp	\Pi
xvp	\xvp	\varpi
xph	\xph	\phi
xcph	\xcph	\Phi
xvph	\xvph	\varphi

xps	\xps	\psi
xcps	\xcps	\Psi
xs	\xs	\sigma
xcs	\xcs	\Sigma
xvs	\xvs	\varsigma
xz	\xz	\zeta
xr	\xr	\rho
xvr	\xvr	\varrho
xt	\xt	\tau
xth	\xth	\theta
xcth	\xcth	\Theta
xvth	\xvth	\vartheta
xu	\xu	\upsilon
xcu	\xcu	\Upsilon
xx	\xx	\xi
xcx	\xcx	\Xi
tex	\tex	\TeX
		\TeX
texs	\texs	\TeX\
		\TeX\
	\bsk	\bigskip
		\bigskip$\mathcal{R}$
		\hskip
		\indent
		\input
	\msk	\medskip
		\medskip$\mathcal{R}$
		\noindent
		\par
	\ssk	\smallskip$\mathcal{R}$
		\smallskip$\mathcal{R}$
		\scriptsize
		\vskip

Keywords defined in `tw-latex.txt`:

-		\textendash
-		\textendash\
--		\textemdash
--		\textemdash\
		\author{$\mathcal{I}$}$\mathcal{R}$
		\begin{
adlen	\adl	\addtolength{$\mathcal{I}$}{}$\mathcal{R}$
		\addtolength{$\mathcal{I}$}{}

adcount	\adc	\addtocounter{$\mathcal{I}$}{}$\mathcal{R}$
		\addtocounter{$\mathcal{I}$}{}
barr		\begin{array}$\mathcal{RIR}$\end{array}
babs	\babs	\begin{abstract}$\mathcal{RIR}$\end{abstract}
bali	\bali	\begin{align}$\mathcal{RIR}$\end{align}
balis	\balis	\begin{align*}$\mathcal{RIR}$\end{align*}
baliat	\baliat	\begin{alignat}{$\mathcal{I}$}$\mathcal{RR}$\end{alignat}
baliats	\baliats	\begin{alignat*}{$\mathcal{I}$}$\mathcal{RR}$\end{alignat*}
balied	\balied	\begin{aligned}{$\mathcal{I}$}$\mathcal{RR}$\end{aligned}
baliedat	\baliedat	\begin{alignedat}$\mathcal{RIR}$\end{alignedat}
baliedato	\baliedato	\begin{alignedat}[$\mathcal{I}$]$\mathcal{RR}$\end{alignedat}
bapp	\bapp	\begin{appendix}$\mathcal{RIR}$\end{appendix}
bbmat	\bbmat	\begin{bmatrix}$\mathcal{RIR}$\end{bmatrix}
bcase	\bcase	\begin{cases}$\mathcal{RIR}$\end{cases}
bcent	\bcent	\begin{center}$\mathcal{RIR}$\end{center}
bcenum	\bcenum	\begin{compactenum}$\mathcal{R}$\item$\mathcal{RIR}$\end{compactenum}
bcenumo	\bcenumo	\begin{compactenum}[$\mathcal{I}$]R\item$\mathcal{RR}$\end{compactenum}
bcitem	\bcitem	\begin{compactitem}$\mathcal{R}$\item$\mathcal{RIR}$\end{compactitem}
bcitemo	\bcitemo	\begin{compactitem}[$\mathcal{I}$]R\item$\mathcal{RR}$\end{compactitem}
bdes	\bdes	\begin{description}$\mathcal{R}$\item[$\mathcal{I}$]$\mathcal{RR}$\end{description}
benu	\benu	\begin{enumerate}$\mathcal{R}$\item$\mathcal{RIR}$\end{enumerate}
benuo	\benuo	\begin{enumerate}[$\mathcal{I}$]R\item$\mathcal{RR}$\end{enumerate}
bequ	\bequ	\begin{equation}$\mathcal{RIR}$\end{equation}
bequs	\bequs	\begin{equation*}$\mathcal{RIR}$\end{equation*}
beqn	\beqn	\begin{eqnarray}$\mathcal{RIR}$\end{eqnarray}
beqns	\beqns	\begin{eqnarray*}$\mathcal{RIR}$\end{eqnarray*}
bf	\bf	\textbf{$\mathcal{I}$}
bfd	\bfseries	\bfseries
bfig	\bfig	\begin{figure}$\mathcal{RIR}$\end{figure}
bfigo	\bfigo	\begin{figure}[$\mathcal{I}$]$\mathcal{RR}$\end{figure}
bmult		\begin{multiline}$\mathcal{RIR}$\end{multiline}
bflalig	\bflalig	\begin{flalign}$\mathcal{RIR}$\end{flalign}
bflaligs	\bflaligs	\begin{flalign*}$\mathcal{RIR}$\end{flalign*}
bfll	\bfll	\begin{flushleft}$\mathcal{RIR}$\end{flushleft}
bflr	\bflr	\begin{flushright}$\mathcal{RIR}$\end{flushright}
bgath	\bgath	\begin{gather}$\mathcal{RIR}$\end{gather}
bgaths	\bgaths	\begin{gather*}$\mathcal{RIR}$\end{gather*}
bgathed	\bgathed	\begin{gathered}$\mathcal{RIR}$\end{gathered}
bgathedo	\bgathedo	\begin{gathered}[$\mathcal{I}$]$\mathcal{RR}$\end{gathered}
bite	\bite	\begin{itemize}$\mathcal{R}$\item$\mathcal{RIR}$\end{itemize}
biteo	\biteo	\begin{itemize}[$\mathcal{I}$]R\item$\mathcal{RR}$\end{itemize}
biblio		\bibliography{$\mathcal{I}$}
bibstyle	\bibstyle	\bibliographystyle{$\mathcal{I}$}

blett	\blett	\begin{letter}{$\mathcal{I}$}$\mathcal{RR}$\end{letter}
blist	\blist	\begin{list}{$\mathcal{I}$}{}$\mathcal{R}$\item$\mathcal{RR}$\end{list}
bminp	\bminp	\begin{minipage}{$\mathcal{I}$}$\mathcal{RR}$\end{minipage}
bminpo	\bminpo	\begin{minipage}[$\mathcal{I}$]{}$\mathcal{RR}$\end{minipage}
bmult	\bmult	\begin{multiline}$\mathcal{RIR}$\end{multiline}
bmults	\bmults	\begin{multiline*}$\mathcal{RIR}$\end{multiline*}
botr		\bottomrule$\mathcal{R}$
bpict	\bpict	\begin{picture}$\mathcal{RIR}$\end{picture}
bpmat	\bpmat	\begin{pmatrix}$\mathcal{RIR}$\end{pmatrix}
bquot	\bquot	\begin{quotation}$\mathcal{RIR}$\end{quotation}
bquo	\bquo	\begin{quote}$\mathcal{RIR}$\end{quote}
bsplit	\bsplit	\begin{split}$\mathcal{RIR}$\end{split}
bsubeq	\bsubeq	\begin{subequations}$\mathcal{RIR}$\end{subequations}
btab	\btab	\begin{tabular}{$\mathcal{I}$}$\mathcal{RR}$\end{tabular}
btabs	\btabs	\begin{tabular*}{$\mathcal{I}$}{}$\mathcal{RR}$\end{tabular*}
btabx	\btabx	\begin{tabularx}{$\mathcal{I}$}{}$\mathcal{RR}$\end{tabularx}
btabl	\btabl	\begin{table}$\mathcal{RIR}$\end{table}
btablo	\btablo	\begin{table}[$\mathcal{I}$]$\mathcal{RR}$\end{table}
btabls	\btabls	\begin{table*}$\mathcal{RIR}$\end{table*}
btablso	\btablso	\begin{table*}[$\mathcal{I}$]$\mathcal{RR}$\end{table*}
btbl	\btbl	\begin{table}$\mathcal{RIR}$\end{table}
btblo	\btblo	\begin{table}[$\mathcal{I}$]$\mathcal{RR}$\end{table}
btbls	\btbls	\begin{table*}$\mathcal{RIR}$\end{table*}
btblso	\btblso	\begin{table*}[$\mathcal{I}$]$\mathcal{RR}$\end{table*}
btabb	\btabb	\begin{tabbing}$\mathcal{RIR}$\end{tabbing}
bbib	\bbib	\begin{thebibliography}{$\mathcal{I}$}$\mathcal{R}$\bibitem{} $\mathcal{RR}$\end{thebibliography}
bindex	\bindex	\begin{theindex}$\mathcal{RIR}$\end{theindex}
btheo	\btheo	\begin{theorem}$\mathcal{RIR}$\end{theorem}
btitpg	\btitpg	\begin{titlepage}$\mathcal{RIR}$\end{titlepage}
btrivl	\btrivl	\begin{trivlist}$\mathcal{RIR}$\end{trivlist}
bvarw	\bvarw	\begin{varwidth}{$\mathcal{I}$}$\mathcal{RR}$\end{varwidth}
bverb	\bverb	\begin{verbatim}$\mathcal{RIR}$\end{verbatim}
bvers	\bvers	\begin{verse}$\mathcal{RIR}$\end{verse}
bibitem		\bibitem{$\mathcal{I}$}$\mathcal{R}$
bibitemo		\bibitem[$\mathcal{I}$]{}$\mathcal{R}$
		\bottomrule$\mathcal{R}$
		\boxed{$\mathcal{I}$}
center		\centering
chap	\chapter{$\mathcal{I}$}	\chapter{$\mathcal{I}$}$\mathcal{R}$
		\cite{$\mathcal{I}$}
		\citep{$\mathcal{I}$}
		\citet{$\mathcal{I}$}

		\caption{$\mathcal{I}$}$\mathcal{R}$
		\cdots
		\cline{$\mathcal{I}$}
cmidr		\cmidrule(I){}
cmidro		\cmidrule[$\mathcal{I}$](){}
		\date{$\mathcal{I}$}$\mathcal{R}$
dd	\dd	\($\mathcal{I}$\)
		\ddots
		\ddot{$\mathcal{I}$}
		\dddot{$\mathcal{I}$}
		\ddddot{$\mathcal{I}$}
		\documentclass{$\mathcal{I}$}$\mathcal{R}$
		\documentclass[$\mathcal{I}$]{}$\mathcal{R}$
		\dots
		\dotsc
		\dotsb
		\dotsm
		\dotsi
		\dotso
em	\emph{$\mathcal{I}$}	\emph{$\mathcal{I}$}
emd		\em
		\end{$\mathcal{I}$}$\mathcal{R}$
		\eqref{$\mathcal{I}$}
foot	\footnote{$\mathcal{I}$}	\footnote{$\mathcal{I}$}
		\footnotesize
fbox		\fbox{$\mathcal{I}$}
fboxo	\fboxo	\framebox[$\mathcal{I}$]{}
		\framebox[$\mathcal{I}$]{}
fboxoo	\fboxoo	\framebox[$\mathcal{I}$][]{}
		\framebox[$\mathcal{I}$][]{}
		\fboxrule{$\mathcal{I}$}
		\fboxsep{$\mathcal{I}$}
geometry	\geometry{}	\geometry{}
		\hline$\mathcal{R}$
		\hspace{$\mathcal{I}$}
		\hspace*{$\mathcal{I}$}
hw		\headwidth
hw2tw		\setlength{\headwidth}{\textwidth}$\mathcal{R}$
href		\href{$\mathcal{I}$}{}
		\include{$\mathcal{I}$}$\mathcal{R}$
incg		\includegraphics{$\mathcal{I}$}$\mathcal{R}$
incgo		\includegraphics[$\mathcal{I}$]{}$\mathcal{R}$
it		\item$\mathcal{R}\mathcal{I}$

ito		\item[$\mathcal{I}$]$\mathcal{R}$
		\intertext{$\mathcal{I}$}
ti	\ti	\textit{$\mathcal{I}$}
itd	\itshape	\itshape
latex	\latex	\LaTeX
		\LaTeX
latexs	\latexs	\LaTeX\
		\LaTeX\
latexe	\latexe	\LaTeXe
		\LaTeXe
latexes	\latexes	\LaTeXe\
		\LaTeXe\
lbl	\lbl	\label{$\mathcal{I}$}
		\label{$\mathcal{I}$}
		\large
		\Large
		\ldots
listf	\listf	\listoffigures$\mathcal{R}$
		\listoffigures$\mathcal{R}$
listt	\listt	\listoftables$\mathcal{R}$
		\listoftables$\mathcal{R}$
mbf	\mbf	\mathbf{$\mathcal{I}$}
		\mathbf{$\mathcal{I}$}
mrm	\mrm	\mathrm{$\mathcal{I}$}
		\mathrm{$\mathcal{I}$}
mcal	\mcal	\mathcal{$\mathcal{I}$}
		\mathcal{$\mathcal{I}$}
msf	\msf	\mathsf{$\mathcal{I}$}
		\mathsf{$\mathcal{I}$}
mtt	\mtt	\mathtt{$\mathcal{I}$}
		\mathtt{$\mathcal{I}$}
mit	\mit	\mathit{$\mathcal{I}$}
		\mathit{$\mathcal{I}$}
mnorm	\mnorm	\mathnormal{$\mathcal{I}$}
		\mathnormal{$\mathcal{I}$}
mdd		\mdseries
mbox		\mbox{$\mathcal{I}$}
mboxo	\mboxo	\makebox[$\mathcal{I}$]{}
		\makebox[$\mathcal{I}$]{}
mboxoo	\mboxoo	\makebox[$\mathcal{I}$][]{}
		\makebox[$\mathcal{I}$][]{}
mpar	\mpar	\marginpar{$\mathcal{I}$}

		\marginpar{$\mathcal{I}$}
midr		\midrule$\mathcal{R}$
multc	\multc	\multicolumn{$\mathcal{I}$}{}{}
multic	\multic	\multicolumn{$\mathcal{I}$}{}{}
		\multicolumn{$\mathcal{I}$}{}{}
ncol	\ncol	&
ncm		\newcommand{$\mathcal{I}$}{}$\mathcal{R}$
newc		\newcommand{$\mathcal{I}$}{}$\mathcal{R}$
ncmo		\newcommand{$\mathcal{I}$}[]{}$\mathcal{R}$
newco		\newcommand{$\mathcal{I}$}[]{}$\mathcal{R}$
ncmoo		\newcommand{$\mathcal{I}$}[][]{}$\mathcal{R}$
newcoo		\newcommand{$\mathcal{I}$}[][]{}$\mathcal{R}$
nct		\newcolumntype{$\mathcal{I}$}{}
newct		\newcolumntype{$\mathcal{I}$}{}
newpg		\newpage$\mathcal{R}$
npg	\npg	\newpage$\mathcal{R}$
nline		\newline$\mathcal{R}$
newlin		\newline$\mathcal{R}$
nlen		\newlength{$\mathcal{I}$}$\mathcal{R}$
newlen		\newlength{$\mathcal{I}$}$\mathcal{R}$
nenv		\newenvironment{$\mathcal{I}$}{}{}$\mathcal{R}$
newe		\newenvironment{$\mathcal{I}$}{}{}$\mathcal{R}$
nenvo		\newenvironment{$\mathcal{I}$}[]{}{}$\mathcal{R}$
neweo		\newenvironment{$\mathcal{I}$}[]{}{}$\mathcal{R}$
nenvoo		\newenvironment{$\mathcal{I}$}[][]{}{}$\mathcal{R}$
neweoo		\newenvironment{$\mathcal{I}$}[][]{}{}$\mathcal{R}$
newt		\newtheorem{$\mathcal{I}$}{}$\mathcal{R}$
newto		\newtheorem{$\mathcal{I}$}{}[]$\mathcal{R}$
newtoo		\newtheorem{$\mathcal{I}$}[]{}$\mathcal{R}$
		\nocite{$\mathcal{I}$}
		\normalsize
pgref		\pageref{$\mathcal{I}$}
par		\paragraph{$\mathcal{I}$}$\mathcal{R}$
pars		\paragraph*{$\mathcal{I}$}$\mathcal{R}$
paro		\paragraph[$\mathcal{I}$]{}$\mathcal{R}$
		\pagebreak$\mathcal{R}$
pgs		\pagestyle{$\mathcal{I}$}$\mathcal{R}$
parbox		\parbox{$\mathcal{I}$}{}
parboxo		\parbox[$\mathcal{I}$]{}{}
pbox	\pbox	\parbox{$\mathcal{I}$}{}
pboxo	\pboxo	\parbox[$\mathcal{I}$]{}{}
pbox		\pbox{$\mathcal{I}$}{}
ref		\ref{$\mathcal{I}$}

rncm		\renewcommand{$\mathcal{I}$}{}$\mathcal{R}$
rnewc		\renewcommand{$\mathcal{I}$}{}$\mathcal{R}$
rnewco		\renewcommand{$\mathcal{I}$}[]{}$\mathcal{R}$
rncmo		\renewcommand{$\mathcal{I}$}[]{}$\mathcal{R}$
rnewcoo		\renewcommand{$\mathcal{I}$}[][]{}$\mathcal{R}$
rncmoo		\renewcommand{$\mathcal{I}$}[][]{}$\mathcal{R}$
		\rm
rmc		\rmfamily
rbox	\rbox	\raisebox{$\mathcal{I}$}{}
		\raisebox{$\mathcal{I}$}{}
rboxo	\rboxo	\raisebox{$\mathcal{I}$}[]{}
		\raisebox{$\mathcal{I}$}[]{}
rboxoo	\rboxoo	\raisebox{$\mathcal{I}$}[][]{}
		\raisebox{$\mathcal{I}$}[][]{}
		\rule{$\mathcal{I}$}{}
		\rule[$\mathcal{I}$]{}{}
		\sc
		\scshape
sec		\section{$\mathcal{I}$}$\mathcal{R}$
secs		\section*{$\mathcal{I}$}$\mathcal{R}$
seco		\section[$\mathcal{I}$]{}$\mathcal{R}$
		\setlength{$\mathcal{I}$}{}
ssec	\ssec	\subsection{$\mathcal{I}$}$\mathcal{R}$
		\subsection{$\mathcal{I}$}$\mathcal{R}$
ssecs	\ssecs	\subsection*{$\mathcal{I}$}$\mathcal{R}$
		\subsection*{$\mathcal{I}$}$\mathcal{R}$
sseco	\sseco	\subsection[$\mathcal{I}$]{}$\mathcal{R}$
		\subsection[$\mathcal{I}$]{}$\mathcal{R}$
sssec	\sssec	\subsubsection{$\mathcal{I}$}$\mathcal{R}$
		\subsubsection{$\mathcal{I}$}$\mathcal{R}$
sssecs	\sssecs	\subsubsection*{$\mathcal{I}$}$\mathcal{R}$
		\subsubsection*{$\mathcal{I}$}$\mathcal{R}$
ssseco	\ssseco	\subsubsection[$\mathcal{I}$]{}$\mathcal{R}$
		\subsubsection[$\mathcal{I}$][]$\mathcal{R}$
spar	\spar	\subparagraph{$\mathcal{I}$}
		\subparagraph{$\mathcal{I}$}
spars	\spars	\subparagraph*{$\mathcal{I}$}
		\subparagraph*{$\mathcal{I}$}
sparo	\sparo	\subparagraph[$\mathcal{I}$]{}
		\subparagraph[$\mathcal{I}$]{}
stcount		\stepcounter{$\mathcal{I}$}$\mathcal{R}$
sf	\sf	\textsf{$\mathcal{I}$}
sfd	\sffamily	\sffamily

scd		\scshape
sl	\sl	\textsl{$\mathcal{I}$}
		\textsl{$\mathcal{I}$}
sld		\slshape
		\text{$\mathcal{I}$}
		\textbf{$\mathcal{I}$}
		\textsf{$\mathcal{I}$}
sc		\textsc{$\mathcal{I}$}
		\textit{$\mathcal{I}$}
		\textup{$\mathcal{I}$}
tt	\tt	\texttt{$\mathcal{I}$}
ttd		\ttfamily
		\texttt{$\mathcal{I}$}
tw	\tw	\textwidth
		\textwidth
		\thanks{$\mathcal{I}$}$\mathcal{R}$
		\title{$\mathcal{I}$}$\mathcal{R}$
tilde	\tilde	\textasciitilde
topr		\toprule$\mathcal{R}$
toc	\toc	\tableofcontents$\mathcal{R}$
tableofcontents		\tableofcontents$\mathcal{R}$
up	\up	\textup{$\mathcal{I}$}
upd		\upshape
url		\url{$\mathcal{I}$}
usep		\usepackage{$\mathcal{I}$}$\mathcal{R}$
usepo		\usepackage[$\mathcal{I}$]{}$\mathcal{R}$
		\vdots
		\vspace{$\mathcal{I}$}$\mathcal{R}$
		\vspace*{$\mathcal{I}$}$\mathcal{R}$
dxa	\dxa	\(\alpha\)
dxb	\dxb	\(\beta\)
dxch	\dxch	\(\chi\)
dxd	\dxd	\(\delta\)
dxcd	\dxcd	\(\Delta\)
dxe	\dxe	\(\epsilon\)
dxve	\dxve	\(\varepsilon\)
dxet	\dxet	\(\eta\)
dxg	\dxg	\(\gamma\)
dxcg	\dxcg	\(\Gamma\)
dxio	\dxio	\(\iota\)
dxl	\dxl	\(\lambda\)
dxcl	\dxcl	\(\Lambda\)
dxm	\dxm	\(\mu\)

dxn	\dxn	\(\nu\)
dxo	\dxo	\(\omega\)
dxco	\dxco	\(\Omega\)
dxp	\dxp	\(\pi\)
dxcp	\dxcp	\(\Pi\)
dxvp	\dxvp	\(\varpi\)
dxph	\dxph	\(\phi\)
dxcph	\dxcph	\(\Phi\)
dxvph	\dxvph	\(\varphi\)
dxps	\dxps	\(\psi\)
dxcps	\dxcps	\(\Psi\)
dxs	\dxs	\(\sigma\)
	\dxcs	\(\Sigma\)
dxvs	\dxvs	\(\varsigma\)
dxz	\dxz	\(\zeta\)
dxr	\dxr	\(\rho\)
dxvr	\dxvr	\(\varrho\)
dxt	\dxt	\(\tau\)
dxth	\dxth	\(\theta\)
dxcth	\dxcth	\(\Theta\)
dxvth	\dxvth	\(\vartheta\)
dxu	\dxu	\(\upsilon\)
dxcu	\dxcu	\(\Upsilon\)
dxx	\dxx	\(\xi\)
dxcx	\dxcx	\(\Xi\)
sqrt	\sqrt	\sqrt{$\mathcal{I}$}
sqrto	\sqrto	\sqrt[$\mathcal{I}$]{}
frac		\frac{$\mathcal{I}$}{}

{array}$\mathcal{RIR}$\end{array}
{abstract}$\mathcal{RIR}$\end{abstract}
{align}$\mathcal{RIR}$\end{align}
{align*}$\mathcal{RIR}$\end{align*}
{alignat}{$\mathcal{I}$}$\mathcal{RR}$\end{alignat}
{alignat*}{$\mathcal{I}$}$\mathcal{RR}$\end{alignat*}
{aligned}$\mathcal{RIR}$\end{aligned}
{aligned}[$\mathcal{I}$]$\mathcal{RR}$\end{aligned}
{alignedat}{$\mathcal{I}$}$\mathcal{RR}$\end{alignedat}
{appendix}$\mathcal{RIR}$\end{appendix}
{bmatrix}$\mathcal{RIR}$\end{bmatrix}
{compactenum}$\mathcal{R}$\item$\mathcal{RIR}$\end{compactenum}
{compactenum}[$\mathcal{I}$]R\item$\mathcal{RR}$\end{compactenum}
{compactitem}$\mathcal{R}$\item$\mathcal{RIR}$\end{compactitem}

```
{compactitem}[ℐ]R\itemℛℛ\end{compactitem}
{cases}ℛℐℛ\end{cases}
{center}ℛℐℛ\end{center}
{description}ℛ\item[ℐ]ℛℛ\end{description}
{enumerate}ℛ\itemℛℐℛ\end{enumerate}
{enumerate}[ℐ]R\itemℛℛ\end{enumerate}
{equation}ℛℐℛ\end{equation}
{eqnarray}ℛℐℛ\end{eqnarray}
{eqnarray*}ℛℐℛ\end{eqnarray*}
{figure}ℛℐℛ\end{figure}
{figure}[ℐ]ℛℛ\end{figure}
{flalign}ℛℐℛ\end{flalign}
{flalign*}ℛℐℛ\end{flalign*}
{flushleft}ℛℐℛ\end{flushleft}
{flushright}ℛℐℛ\end{flushright}
{gather}ℛℐℛ\end{gather}
{gather*}ℛℐℛ\end{gather*}
{gathered}ℛℐℛ\end{gathered}
{gathered}[ℐ]ℛℛ\end{gathered}
{itemize}ℛ\itemℛℐℛ\end{itemize}
{itemize}[ℐ]R\itemℛℛ\end{itemize}
{letter}{ℐ}ℛℛ\end{letter}
{list}{ℐ}{}ℛ\itemℛℛ\end{list}
{minipage}{ℐ}ℛℛ\end{minipage}
{minipage}[ℐ]{}ℛℛ\end{minipage}
{multiline}ℛℐℛ\end{multiline}
{multiline*}ℛℐℛ\end{multiline*}
{picture}ℛℐℛ\end{picture}
{pmatrix}ℛℐℛ\end{pmatrix}
{quotation}ℛℐℛ\end{quotation}
{quote}ℛℐℛ\end{quote}
{split}ℛℐℛ\end{split}
{subequations}ℛℐℛ\end{subequations}
{tabular}{ℐ}ℛℛ\end{tabular}
{tabular*}{ℐ}{}ℛℛ\end{tabular*}
{tabularx}{ℐ}{}ℛℛ\end{tabularx}
{table}ℛℐℛ\end{table}
{table}[ℐ]ℛℛ\end{table}
{table*}ℛℐℛ\end{table*}
{table*}[ℐ]ℛℛ\end{table*}
{tabbing}ℛℐℛ\end{tabbing}
{thebibliography}ℛℐℛ\end{thebibliography}
{theindex}ℛℐℛ\end{theindex}
```

```
{theorem}RIR\end{theorem}
{titlepage}RIR\end{titlepage}
{trivlist}RIR\end{trivlist}
{varwidth}{I}RR\end{varwidth}
{verbatim}RIR\end{verbatim}
{verse}RIR\end{verse}
```

There are also environment codes (above) without `\begin{` (which is itself a keyword); this allows, if one started to input an environment code, to finish it by `[Tab]`.

5.4 Compiling TeXworks under Linux

From the TeXworks web site.

Requirements

To build the current TeXworks prototype from source, you will need to install developer packages (or equivalent) for:

Qt4(4.3.2 is known to work; later versions should also work. Note that on GNU/Linux, TeXworks may crash when changing the spell-checking language when using Qt 4.3.x. Updating to Qt 4.4.0 solves this.)
`http://trolltech.com/downloads/opensource#qt-open-source-edition`

Poppler (0.8.0 or later recommended, although 0.6 series should work)
`http://poppler.freedesktop.org/`

Hunspell (at least 1.2.2 recommended; not tested with earlier versions)
`http://hunspell.sourceforge.net/`

libdbus (on X11-based platforms only; not used on Mac OS X or Windows)
`http://dbus.freedesktop.org/`

along with their dependencies (such as Freetype,fontconfig, zlib, etc.)

Details will depend on your platform.

On GNU/Linux or similar systems, your package manager can probably provide all these. Feel free to contribute distribution-specific notes for the Building page.

After installing the necessary libraries (and their headers), check the TeXworks.pro file and update paths as necessary; some of the paths currently in the project file are specific to one of my build machines.

Once the project file is correct, run qmake to create a Makefile or Xcode project, and then run make or use Xcode to build the application. Do not run qmake -pro to recreate the .pro file, as it has been hand-edited.

Note: on some systems, the default qmake may be a Qt3 version, in which case it will generate lots of warnings, and make will fail miserably; TeXworks is a Qt4 program. In that case, see if you have qmake-qt4, or perhaps run the Qt4 version using a full path.

The current version of TeXworks has been successfully built with

- Xcode (using gcc 4) on Mac OS X 10.4 and 10.5
- gcc 4 on GNU/Linux

- a combination of mingw32 on Windows XP and the mingw32 cross-compiler hosted on Mac OS X

Getting all the right libraries built and installed for the mingw32 build can be tricky. By slightly patching poppler 0.8.0, it is possible to build a version that does not depend on fontconfig (and thence on various other libraries), though it will then not work properly for PDFs with non-embedded fonts.

To avoid having to build the poppler dependencies for Windows, you can get precompiled libraries from the KDE on Windows project.

Alternatively, I believe it would also be possible to build with Visual C, but I have not done this personally.

Notes on building TeXworks on various distributions.

Building TeXworks on typical GNU/Linux systems should be fairly straightforward. You just need standard build tools (gcc, make, etc.), and the development files (not just runtime libraries) for Qt4, poppler,hunspell, and dbus. The exact set of packages needed will depend how your distribution arranges things; some examples are given here.

Fedora 9

```
# yum install poppler-devel qt4-devel hunspell-devel
$ svn checkout http://texworks.googlecode.com/svn/trunk/ texworks-read-only
$ cd texworks-read-only
$ qmake
$ make
$ ./texworks
```

(Thanks to Dave Crossland for this.)

Ubuntu (Hardy) & Debian

```
$ sudo aptitude install build-essential libpoppler-qt4-dev libhunspell-dev libdbus-1-dev
$ svn checkout http://texworks.googlecode.com/svn/trunk/ texworks-read-only
$ cd texworks-read-only
$ qmake-qt4
$ make
$ ./texworks
```

(I hope this is OK; have not checked on a completely virgin Ubuntu system, so it is possible I missed something in the dependencies.)

Another option would be to use the Ubuntu package announced by Martin Sievers, thus avoiding the need to compile from source.

References

[1] **D. Knuth**, *The TEXbook*, Addison Wesley, 1986-1992

[2] **D. Knuth**, *The METAFONT book*, Addison Wesley, 1986-1992

[3] **L. Lamport**, *LATEX, user's guide and reference manual*, Addison Wesley, 1985 (LATEX2.09), 1994 (LATEX2ϵ)

[4] **M. Goossens, F. Mittelbach & A. Samarin**, *The LATEX Companion*, Addison Wesley, 1994